VISUAL BASIC

ศรันย์ ไมตรีเวช

ประโยชน์ที่ชัดเจนของ วิชวลเบสิก 4.0

น้ำหนักชนิดใดบ้างที่ทำให้เกิดแนวโน้มในการตัดสินใจลงทุนลงแรงกับวิชวลเบสิก 4.0

มีอยู่หลายเหตุผลเหลือเกินครับที่ทำให้องค์กรต่าง ๆ รวมทั้งโปรแกรมเมอร์หลายคนรู้สึกลังเลกับการทำงานภายใต้สภาพแวดล้อมของวินโดวส์ 95 อาทิเช่น ได้ยินกิตติศัพท์เกี่ยวกับข้อผิดพลาดของโอเอสตัวนี้จนหูฉี่ หรือไม่ก็มีข่าวที่ค่อนข้างน่าเชื่อว่าวินโดวส์ 95 จะถูกโละทิ้งในเร็ววัน ควรรอโอเอสที่กำลังจะมาแรงและน่าเชื่อถือกว่าอย่างวินโดวส์เอ็นที 4.0 รวมทั้งการปรากฏตัวอย่างเหลือเชื่อของวินโดวส์ 96 เล่นเอาเราท่านถึงกับช็อคราวกับโดนไฟดูด เพราะบางคนยังไม่กล้าเสี่ยงลงวินโดวส์ 95 เพื่อใช้งานจริงจังในเครื่องของตัวเองด้วยซ้ำ

หลายคนเกิดความรู้สึกคล้ายโดนไมโครซอฟต์หลอกให้เข้ามาเล่นขายของ น่าอัศจรรย์ที่แผนการโปรโมทวินโดวส์ 95 ของไมโครซอฟต์ยิ่งใหญ่เสียจนคนทั้งโลกหลงคอยว่าจะเป็นความหวังใหม่ แต่แล้วเมื่อสินค้าออกตัวมาจริงสู่มือผู้ใช้นับล้าน ๆ คน ผลสะท้อนจากผู้ใช้ตัวจริงที่ไม่ใช่สื่อช่วยประโคมกลับติดลบแบบจมธรณี เท่าที่ผมคุยกับผู้ใช้และโปรแกรมเมอร์ทั้งประเทศไทยและต่างประเทศ จะได้ยินอะไร ๆ ที่สรุปได้กระชับหน่อยว่าวินโดวส์ 95 "ดีจริง เร็วจริง และเฮงซวยจริงๆ" ว่าไปแล้วก็คือมีครบทุกรสครับ เปรี้ยว หวาน มัน เค็ม เผ็ด จืด สมควรจะเป็นซอฟต์แวร์แห่งการกล่าวขานถึงจริง ๆ

ไม่ว่าจะเป็นการเปลี่ยนแปลงแนวการใช้วินโดวส์ หรือแอพพลิเคชันต่าง ๆ อย่างเช่นวิชวลเบสิก ค่อนข้างแสดงให้เห็นชัดว่าทีมงานออกแบบของไมโครซอฟต์ยังทำต้นแบบไม่ลงตัว หากเป็นผู้ใช้คอมพิวเตอร์มืออาชีพก็คงไม่หนักแรงกับการติดตามความเปลี่ยนแปลงเคลื่อนไหวสักเท่าไหร่ แต่ถ้าคิดในแง่ผู้ใช้ทั่วไปซึ่งพื้นฐานไม่ได้บังคับให้เขาต้อง "เก่ง" คอมพิวเตอร์แล้ว ก็นับว่าไมโครซอฟต์ (ซึ่งปัจจุบันเป็นเจ้าใหญ่นายโตในการกำหนดทิศทางซอฟต์แวร์สำหรับผู้ใช้ส่วนมาก) ได้ทำทารุณกรรมกับผู้ใช้อยู่มากโข อันนี้น่าจะถือเป็นความผิดขั้นอุกฉกรรจ์ประการแรก

เท่าที่ผมเดิน ๆ สุ่มไปในที่ต่าง ๆ ผมคงพบเหมือนกับคุณหลาย ๆ คน ว่าหน้าจอคอมพิวเตอร์ในเมืองไทยยังเป็นวินโดวส์ 3.11 นานครั้งหรอกครับถึงจะเห็นวินโดวส์ 95 ซึ่งก็มักจะเป็นเครื่องของโปรแกรมเมอร์มากกว่าผู้ใช้ธรรมดา เครื่องของผมเองมีทั้งสองแพลตฟอร์มนั่นแหละ เพราะเคยเจออะไรเด็ด ๆ มาแล้วหลายหน ผมจึงอยากจะพร้อมย้ายการทำงานทันทีที่เกิดความไม่เข้ากันระหว่างแอพพลิเคชันเก่ากับแพลตฟอร์มใหม่อย่างวินโดวส์ 95

และโดยการใช้งานทุกวันนี้ของผม หากไม่จำเป็นแล้วผมจะไม่ค่อยแอพพลิเคชันบนวินโดวส์ 95 เท่าไหร่ เหตุหนึ่งอาจเป็นเพราะแอพพลิเคชันประจำวันจริง ๆ ของผมเป็นของเก่า 16 บิต ซึ่งผมไม่ค่อยจะไว้ใจวินโดวส์ 95 เท่าไหร่นักว่ามันสามารถดูแลแอพพลิเคชันเหล่านั้นได้ลงตัวเพียงใด โดยเฉพาะอย่างยิ่งแอพพลิเคชันที่ต้องมีการจัดเก็บไฟล์แล้วดึงมาใช้ใหม่หรือส่งต่อไปให้คนอื่น ไฟล์ที่จัดเก็บไว้นำไปให้นักพนันเล่นกันได้เลยครับ เพราะเมื่อ

ย้อนไปใช้กับแพลตฟอร์ม 16 บิตแล้วมีโอกาสเท่ากันครึ่งต่อครึ่งระหว่างใช้ได้กับคว้าน้ำเหลว

แล้วผมก็ไม่ค่อยชอบวิธีการปัดความรับผิดชอบของไมโครซอฟต์ หรือความพยายามบ่ายเบี่ยงเลี่ยงโน่นเลี่ยงนี่ว่ามันไม่ใช่ความผิดของพวกเขาทั้งหมด อ้างว่าผู้ใช้ไม่ศึกษาให้กระจ่างถึงการใช้และวิธีแก้ปัญหา (ทั้งที่ก่อนออกตัวคุยโขมงโฉงเฉงเลยว่าวินโดวส์ 95 ใช้ง่ายที่สุดในโลก เหนือวินโดวส์ทุกเวอร์ชันก่อนหน้า) หรือไม่ก็พยายามสร้างภาพพจน์ขอความเห็นใจ ถึงกับใช้พื้นที่อันมีค่าในซอฟต์แวร์เอ็นไซโคลปีเดียของตัวเองอย่าง Encarta มากล่าวถึงความล่าช้าของการออกวางตลาดว่าเป็นเพราะโปรแกรมใหญ่มหึมา (วินโดวส์ 3.x คอมพายล์โค้ดไปทั้งหมด 3 ล้านบรรทัด ส่วนวินโดวส์ 95 ว่าเข้าไป 11 ล้านบรรทัด)

ประเด็นมันไม่น่ามาอ้างถึงความซับซ้อน ชวนให้เห็นใจอย่างโน้นอย่างนี้ ในเมื่อตัวก็คิดเอง เสนอเอง และทำเองทั้งหมด ไม่มีใครเขาไปแทรกแซงที่ตรงไหน ฉะนั้นเมื่อริเริ่มสร้างแล้วก็ต้องสานให้เสร็จสมบูรณ์ตามหน้าที่ของแต่ละฝ่าย สำหรับผู้ใช้แล้วมีหน้าที่เป็นฝ่ายซื้อสินค้าซึ่งไม่สมควรรับผิดชอบอะไรกับกลไกความยุ่งยากภายในของกระบวนการผลิตสินค้า ทำนองเดียวกันกับที่ผู้ซื้อเสื้อผ้าใส่เขาจะหวังความสวยงามคงทนของเนื้อผ้า ไม่ต้องรู้เห็นว่ามันผ่านกระบวนการผลิตที่พิสดารพันลึกมากเพียงใด หากปรากฏว่าใส่ทีเดียว พอซักตากแห้งเท่านั้น มาใส่อีกทีเห็นเสื้อแสงขาดกระจุย อย่างนี้ความผิดก็สมควรตกอยู่กับโรงงานทอผ้าลูกเดียว ไม่ใช่มาโทษผงซักฟอกหรือวิธีซักของผู้ใช้ไม่ตรงตามเงื่อนไขของตน

สิ่งที่น่ากล่าวโทษไมโครซอฟต์มากกว่าเรื่องอื่นใดทั้งหมดเห็นจะได้แก่ความเข้ากันได้ระหว่างแอพพลิเคชัน 16 บิต กับแพลตฟอร์ม 32 บิต ไมโครซอฟต์พยายามจูงใจให้เชื่อว่าตนให้ความสำคัญและทุ่มเทกำลังสมองอย่างถึงที่สุดเพื่อทำแพลตฟอร์มที่ดูแลดำเนินการกับแอพพลิเคชันทั้ง 16 บิต และ 32 บิต แต่เอาเข้าจริงก็ได้ผลแบบที่ทำให้ชาวบ้านต้องเชื่อครึ่งไม่เชื่อครึ่ง เนื่องจากเกือบร้อยจากร้อยที่ใช้วินโดวส์ 95 รันแอพพลิเคชันเก่าได้เพียงห้านาทีหรืออย่างเก่งครึ่งชั่วโมง เป็นต้องร้อง "ว้า..." หรือ "ยี้..." ออกมายาว ๆ กันทั้งนั้น

ช่วงกลางปี 1995 ไมโครซอฟต์มีความพยายามอย่างยิ่งที่จะทำลิสต์ไฟล์เก่า 16 บิต ที่เข้ากันได้กับวินโดวส์ 95 ออกมา เพื่อให้ผู้ใช้สบายใจว่าแอพพลิเคชันยอดนิยมเหล่านั้นใช้แล้ว "ไม่มีปัญหา" แต่ที่ผมเจอและลองกับตัวเอง บางแอพพลิเคชันในลิสต์ที่ไม่มีปัญหานั้นความจริงมีเสียยิ่งกว่ามี ภาพพจน์อันน่าเชื่อถือของไมโครซอฟต์ที่เคยมีอยู่ในใจผมแทบล้มละลายตายจาก เพราะมันแสดงให้เห็นถึงความพยายามโปรโมทแบบสุดตัวอย่างเดียว ต้องการสร้างภาพพจน์ด้วยลมปากลูกเดียวราวกับนักการเมือง ไม่คำนึงแม้แต่น้อยว่าเมื่อผู้ใช้เป็นล้าน ๆ คน เขาพบความจริงแล้วจะรู้สึกอย่างไร

หากความไม่น่าเชื่อถือดำเนินต่อไปเช่นนี้ ก็คงมีผลให้ผู้ใช้ส่วนใหญ่ต้องพิจารณาต่อว่าหากจะติดตามเทคโนโลยีของไมโครซอฟต์ตามวันและเวลาปัจจุบัน จะเป็นเรื่องเสี่ยงสำหรับแอพพลิเคชันเก่าซึ่งเคยใช้มาและยังไม่พร้อมจะปรับปรุงใหม่เป็น 32 บิตหรือไม่?

ด้วยประสบการณ์ตรงและความโน้มเอียงของผู้ใช้ทั่วไปดังกล่าวมาแล้วข้างต้นนี้ มองเผิน ๆ เบื้องแรกคงเห็นได้ว่าการพัฒนาแอพพลิเคชันด้วยวิชวลเบสิก 4.0 หรือเครื่องมือพัฒนาแอพพลิเคชัน 32 บิตใด ๆ อาจยังไม่จำเป็นถึงขนาดเสียเปรียบคู่แข่งหากคุณยังไม่กระโดดลงสนาม เนื่องจากเมื่อพัฒนาแอพพลิเคชัน 32 บิตออกมาแล้ว ก็ไม่แน่ว่าจะมีใครเขาพร้อมจะใช้ (นี่พูดกว้าง ๆ เฉพาะที่เห็นในเมืองไทยนะครับ หากคุณเข้าอินเทอร์เน็ตและแลกเปลี่ยนความคิดเห็นกับคนส่วนใหญ่ โดยเฉพาะในอเมริกา จะเห็นว่ามีคนยอมรับวินโดวส์ 95 ไว้ใช้จริงมากพอควร และการสร้างแอพพลิเคชันด้วยวิชวลเบสิก 4.0 ก็เป็นเรื่องใหญ่พอจะมีการระดมโปรแกรมเมอร์ผู้มีความสามารถใช้เทคโนโลยีใหม่ชิ้นนี้ด้วยอัตราค่าจ้างที่สูงลิบ)

วันเวลาจะไม่หยุดอยู่แค่วันนี้กับอาทิตย์ที่แล้ว เชื่อได้เลยครับว่าคุณเผลอไม่ดูพระอาทิตย์ตกแค่ร้อยหรือสองร้อยหน การใช้คอมพิวเตอร์ทั่วโลกจะเคลื่อนย้ายมาอยู่บนแพลตฟอร์มที่เป็น 32 บิตกันหมด ทั้งฮาร์ดแวร์และซอฟต์แวร์ใหม่จะค่อย ๆ ทยอยกันทอดทิ้งความเป็น 16 บิตไว้กับโลกมืดหลังเขา

ฉะนั้น สำหรับโปรแกรมเมอร์เราท่าน คำถามที่ว่าการพัฒนาแอพพลิเคชัน 32 บิต มีความจำเป็นไหม หากมองไปยาว ๆ ข้างหน้าคำตอบคือจำเป็นอย่างยิ่ง

ในเมื่อหลีกเลี่ยงไม่ได้ที่จะต้องยอมรับเทคโนโลยี 32 บิตเสียแต่วันนี้ คำถามที่น่าสนใจต่อไปก็คือ เครื่องมือพัฒนาแอพพลิเคชันอย่างไมโครซอฟต์วิชวลเบสิกนั้นเก่งพอที่จะให้คุณคอมพายล์โค้ดให้เป็นได้ทั้งแอพพลิเคชัน 16 บิต และ 32 บิตหรือไม่ (คุณจะได้ใช้โค้ดเดียวกันสร้าง EXE ไฟล์ไปรันบนทุกแพลตฟอร์มอย่างไร้กังวล) ลองคิดดูว่าผู้ใช้จะประทับใจงานของคุณเพียงใด หากเขาบอกว่าวันนี้ขอรันบนวินโดวส์ 3.11 ไปก่อน แล้วอีกปีหนึ่งข้างหน้าเขาไหว้วานให้คุณเปลี่ยนแอพพลิเคชันของเขามารันบนวินโดวส์ 9x (ซึ่งเขาจะเห็นว่าการประมวลผลเร็วขึ้นผิดหูผิดตา) คุณก็สามารถทำให้เขาได้โดยไม่มาโอดกาเหว่าขอค่าเหนื่อยเพิ่ม และการคอมพายล์จะเป็นไปอย่างรวดเร็วภายในนาทีเดียวเสียด้วย ตกลงไม่มีใครต้องเหนื่อย ไม่มีใครต้องรอ

วิชวลเบสิกสามารถคอมพายล์โค้ดของคุณให้เป็น 16 บิต หรือ 32 บิตก็ได้ตามต้องการ ผลดีประการแรกที่เห็นได้อย่างชัดเจนน่าจะได้แก่ คุณสามารถทำงานในสภาพแวดล้อมที่รวดเร็ว เต็มไปด้วยความฉับไวทุกประการ ในขณะเดียวกันก็ไม่ต้องเป็นกังวลกับการมีหรือไม่มีผู้ใช้รองรับ ถ้าเขายังสมัครใจผูกติดอยู่กับวินโดวส์ 3.x ก็คอมพายล์เป็น 16 บิตให้เขา ถ้าเขาชอบตามเทคโนโลยี ลงวินโดวส์ 95 ไว้ในเครื่องแล้วก็อย่าให้เสียประโยชน์ คอมพายล์เป็น 32 บิตให้เขา เพื่อความรวดเร็วที่เถียงไม่ได้

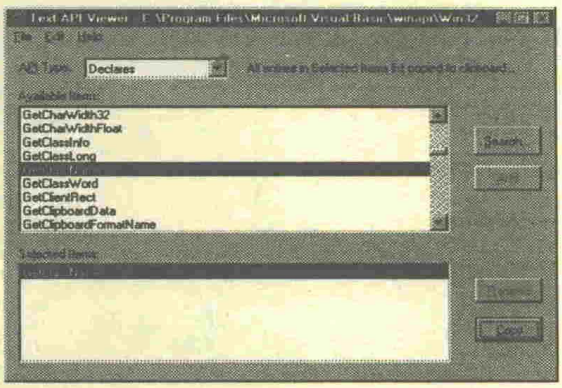
บางคนอาจยังงงว่าการคอมพายล์ให้เป็น 32 บิต และ 16 บิต นั้น ทำได้ในสภาพแวดล้อมเดียวหรือไม่ เช่น คุณเล่นวิชวลเบสิก 4.0 บนวินโดวส์ 95 จะสามารถคอมพายล์ให้รันเป็น 16 บิตได้หรือเปล่า อันนี้ต้องทำความเข้าใจให้ดีว่าไม่ได้นะครับ หากผู้ใช้เขาต้องการเวอร์ชัน 16 บิต ก่อนอื่นคุณต้องลงวิชวลเบสิก 4.0 ภายใต้สภาพแวดล้อมของวินโดวส์ 3.x แล้วจึงทำการคอมพายล์ให้เป็น 16 บิตตามต้องการ สิ่งที่คุณไม่ต้องเหนื่อยคือการนั่งหลังขดหลังแข็งเขียนโค้ดใหม่เท่านั้น

มีความไม่เข้ากันบางประการระหว่างแพลตฟอร์ม 16 บิต กับ 32 บิต โดยเฉพาะอย่างยิ่งที่เกี่ยวกับฟังก์ชัน API อันนี้คุณต้องรู้ให้แน่ ๆ ว่าฟังก์ชันที่ไม่เข้ากันเหล่านั้นมีอะไรบ้าง และต้องใช้ในแอพพลิเคชันของคุณหรือเปล่า เมื่อรู้แล้วคุณต้องเขียนโค้ดลงไปตรง ๆ ครับ ไม่มีวิธีเลี่ยงอื่น และต่อไปนี้เป็นการเขียนโค้ดเพื่อให้เกิดการคอมพายล์โดยการตรวจเช็คว่าแพลตฟอร์มที่ทำการคอมพายล์นั้นเป็น 16 บิต หรือ 32 บิต

การประกาศฟังก์ชัน

ที่ว่าฟังก์ชัน API มันต่างกันนั้นต่างกันตรงไหน? มันต่างกันตรงชื่อไลบรารีและชนิดข้อมูลที่คุณใช้เวลาประกาศฟังก์ชัน เช่น ฟังก์ชัน GetClassName อย่างนี้ ไลบรารีของ 32 บิต จะเป็น User32 ส่วน 16 บิต จะเป็น User เฉย ๆ

คุณอาจจะนึกขึ้นมาในใจว่า อย่างนี้จะไปรู้ได้ยังไงล่ะว่าฟังก์ชัน API เก่าที่เคยใช้มาก่อนจะมีอะไรบ้างที่ต้องเปลี่ยนแปลง อันนี้ง่ายนิดเดียวครับ หากเคยคุ้นกับการใช้ฟังก์ชันไหน อย่าเพิ่งด่วนเขียนมันลงไปในโค้ด (ปกติเราก็ไม่ใช้วิธีเขียนจากความจำอยู่แล้วจริงไหมครับ?) ให้ลองดูไฟล์ WIN32API.TXT ด้วยโปรแกรม API Viewer ซึ่งมาพร้อมกับวิชวลเบสิก 4.0 เสียก่อน (เท็กซ์ไฟล์สามารถเปลี่ยนให้เป็นดาต้าเบสได้ด้วยเมนูคำสั่ง FileConvert Text to Database ซึ่งอยู่ภายในตัวโปรแกรมเอง เพื่อความรวดเร็วในการค้นหายิ่งขึ้น)



รูปที่ 1 การใช้ API Viewer ดูรายละเอียดของการประกาศตัวแปรฟังก์ชัน API

ในรูปที่ 1 คือสิ่งที่เราจะเห็นหลังจากโหลดไฟล์ WIN32-API.TXT ขึ้นมาแล้ว (ด้วยเมนูคำสั่ง FileLoad Text File) คุณสามารถใช้ปุ่ม Search หรือพิมพ์ชื่อฟังก์ชันมากอักษรเท่าที่คุณจำได้ลงไปขณะโฟกัสที่ลิสต์ Available Items คุณจะเห็นชื่อฟังก์ชัน API ที่ต้องการถูกโฟกัส ตอนนั้นถ้าหากคุณดับเบิลคลิ๊กเลือก หรือคลิ๊กปุ่ม Add จะเห็นในช่อง Selected Item ปรากฏชื่อฟังก์ชัน API นั้น ๆ

โอกาสนี้แหละครับ หากคุณคลิ๊กปุ่ม Copy ก็จะมีรายละเอียดการประกาศฟังก์ชัน API เข้าไปอยู่ในคลิปบอร์ด ดังนั้นคุณจึงสามารถ paste มาลงในส่วนประกาศฟังก์ชันใต้สภาพแวดล้อมของวิชวลเบสิกทันที ดังจะเห็นในกรณีของ GetClassName เป็นดังนี้

```
Declare Function GetClassName Lib "user32"
Alias "GetClassNameA" (ByVal hwnd As Long,
ByVal lpClassName As String, ByVal nMaxCount As
Long) As Long
```

แต่เมื่อคุณใช้ API Viewer โหลดไฟล์ WIN31API.TXT ซึ่งจะให้มากับเวอร์ชัน 16 บิต คุณจะพบว่ารายละเอียดในการประกาศมันต่างไปเป็น

```
Declare Function GetClassName Lib "User" Alias
"GetClassNameA" (ByVal hWnd As Integer, ByVal
lpClassName As String, ByVal nMaxCount As
Integer) As Integer
```

เราต้องเลือกประกาศเพียงแบบใดแบบหนึ่ง จึงต้องตั้งเงื่อนไขสำรวจดูว่าโค้ดกำลังถูกคอมพายล์ด้วยคอมพายเลอร์ 16 บิต หรือว่า 32 บิต โดยใช้ไดเรคทีฟ #If ... Then ... #Else เข้ามาช่วย ที่ไม่สามารถใช้โครงสร้าง If ... Then ... Else ตามปกติก็เพราะเราต้องไปอยู่ในส่วนประกาศตัวแปรซึ่งเขียนโค้ดไม่ได้ สำหรับการใช้ไดเรคทีฟดังกล่าวจะมีซินแทกซ์ดังนี้

```
#If นิพจน์ Then
        คำสั่ง
[#ElseIf นิพจน์ Then
        [คำสั่ง]]
[#Else
        [คำสั่ง]]
#End If
```

ขั้นต่อไปคือเราต้องทราบว่าจะใช้ค่าคงที่ (constant) ใดมาเป็นตัวตัดสิน ว่าคอมพายเลอร์เป็น 16 บิต หรือ 32 บิต มีค่าคงที่อยู่สองตัวครับคือ Win32 และ Win16 ทั้งสองตัวนี้มีฐานะเป็นบูลีน หากคืนค่ามาเป็น True ก็แสดงว่าคอมพายเลอร์เป็นเวอร์ชันนั้น ๆ

จากหลักการทั้งหมดนี้เราจะเขียนโค้ดว่า

```
# If Win32 Then
Declare Function GetClassName Lib "user32" Alias
"GetClassNameA" (ByVal hwnd As Long, ByVal
lpClassName As String, ByVal nMaxCount As Long)
As Long
# ElseIf Win16 Then
Declare Function GetClassName Lib "User" Alias
"GetClassNameA" (ByVal hWnd As Integer, ByVal
lpClassName As String, ByVal nMaxCount As
Integer) As Integer
# End If
```

เมื่อรู้ว่าสามารถคอมพายล์กลับไปกลับมาระหว่างเวอร์ชัน 16 บิต กับ 32 บิตอย่างนี้แล้ว คุณก็คงไม่เป็นกังวลอีกต่อไปเกี่ยวกับการเขียนโค้ดว่าจะเกิดความขัดแย้งใดขึ้นหรือเปล่า หากคุณยังไม่พร้อมจะลงวินโดวส์ 95 ด้วยเหตุผลของความกลัวข้อผิดพลาด หรือไม่พร้อมจะอัพเกรดเครื่อง หรือเหตุผลแวดล้อมร้อยแปดพันเก้าอื่นใดก็สุดแล้วแต่ ก็มั่นใจได้ว่าโค้ดจะไม่ล้าสมัย

ในขณะนี้ผมเชื่อ (ไม่จำเป็นก็ไม่ต้องเชื่อตามผมนะครับ) ว่าวิชวลเบสิกพัฒนามาจนถึงขั้นอยู่ตัว เริ่มเข้าที่คงเส้นคงวาแล้ว ดูจากที่ไมโครซอฟต์บรรลุเป้าหมายที่เคยวางไว้ นั่นคือ รวมเอาแอพพลิเคชันของตนทั้งหมดเข้าเป็นหนึ่ง โดยมีภาษาวิชวลเบสิกมาช่วยเป็นแมคโครให้ได้สำเร็จ หากวิชวลเบสิกจะพัฒนาขั้นต่อ ๆ ไปก็คงไม่รื้อโครงสร้างสภาพแวดล้อมเหมือนอย่างที่เห็นเวอร์ชัน 3.0 กระโดดมาเป็น 4.0 การพัฒนาคงเกี่ยวข้องกับเรื่องของความเป็น OOP หรือไม่ก็ไคล์เอ็นต์เซอร์ฟเวอร์มากกว่า

หากเป็นไปตามที่ผมเชื่อนี้ ก็หมายความว่า โค้ดของคุณที่เขียนขึ้นด้วยวิชวลเบสิก 4.0 ในวันนี้ จะไม่ล้าสมัยไปอีกอย่างน้อยก็ 3 - 4 ปี หรือกว่านั้น วิสัยทัศน์ของผมไม่ยาวเหมือนพี่บิล เกตส์แก ก็เอาเป็นว่าผมประมาณเอาด้วยวิธีของหมอเดา (ญาติห่าง ๆ ของหมอดู) อย่างนี้ก็แล้วกัน

ประโยชน์ที่คุณจะได้รับจากวิชวลเบสิก 4.0 ซึ่งผมเห็นว่าสลักสำคัญเป็นอันมากอีกข้อหนึ่งเห็นจะได้แก่ ฟีเจอร์ที่ทำให้พวกเราเขียนโค้ดได้เก่งขึ้น ใช้สอยองค์ประกอบที่มีอยู่ในวิชวลเบสิกอย่างเต็มเม็ดเต็มหน่วย โดยไม่ต้องเปลืองแรงเปลืองเวลาศึกษาวิชวลเบสิกอย่างละเอียดถี่ถ้วนเสียก่อน

สมัยนี้ข้อมูลคืออำนาจจริง ๆ ครับ และถ้าใครสามารถกำข้อมูลไว้ได้ในมือมากเพียงใด โอกาสที่จะได้เปรียบหรือสร้างผลงานที่ดีกว่าตัวเองเมื่อวันวานก็เป็นไปได้สูง การที่จะมีข้อมูลอยู่ในมือคราวเดียวกันทีละมาก ๆ นั้น ทางที่เป็นไปได้คือต้องบีบอัดข้อมูลแต่ละตัวให้เล็กที่สุด กะทัดรัดชัดเจนและได้ใจความมากที่สุด มิฉะนั้นก็ยากเหลือเกินที่สำพังสมองมนุษย์ขนาดส้มโอใบเขื่องนี่จะไปรองรับอะไรต่อมิอะไรได้หมด

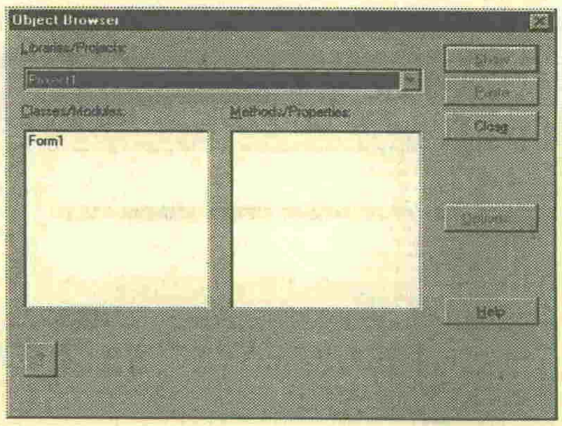
คงเป็นการดีถ้าคุณมีข้อมูลเกี่ยวกับออปเจ็กต์ทั้งหมดในอาณาจักรซอฟต์แวร์ของไมโครซอฟต์ไว้ในมือ โดยที่ข้อมูลทุกชิ้นเล็กกะทัดรัดอ่านง่าย หาก็ง่าย และนั่นไม่ใช่ฝันที่เกินเอื้อมอีกต่อไปเมื่อวิชวลเบสิก 4.0 มอบออปเจ็กต์บราวเซอร์ให้พวกเราเป็นกำนัล

ออปเจ็กต์บราวเซอร์

สำหรับหน้าต่างออปเจ็กต์บราวเซอร์นี้ หากไม่ได้ลองใช้วิชวลเบสิก 4.0 จริง ๆ ก็คงยังมองไม่ออกว่ามันจะมีประโยชน์สักเพียงใด หรือถ้าใครลองใช้แล้วไม่เข้าไปดูในรายละเอียดดี ๆ บางทีก็อาจพลาดฟีเจอร์สำคัญบางจุดไป ผมจึงขอนำเนื้อหาสำคัญมากล่าวถึงทั้งหมดในที่นี้ เพื่อที่จะรับรองว่าการเคลื่อนย้ายจากเวอร์ชัน 3.0 ขึ้นมาเป็น 4.0 นั้นคุ้มค่าเพียงพอ อย่างน้อยก็ในแง่ของวิธีให้ความรู้ในการประยุกต์องค์ประกอบของซอฟต์แวร์ไมโครซอฟต์เพื่อสร้างแอพพลิเคชันของคุณ ออปเจ็กต์บราวเซอร์มีส่วนช่วยให้เกิดความมั่นใจในการสร้างงานที่ทรงประสิทธิภาพเต็มอัตรา

หน้าต่างออปเจ็กต์บราวเซอร์มีไว้ทำไม? หน้าที่ของมันก็ตรงตามชื่อเลยครับ นั่นคือเป็นตัวสำรวจรายละเอียดของออปเจ็กต์ กล่าวคือมันจะแสดงให้คุณเห็นคลาส พรอพเพอร์ตี้ส์ และเมธอดของออปเจ็กต์หนึ่ง ๆ นอกจากนี้ยังเปิดโอกาสให้คุณบันทึกเป็นไทยหรืออังกฤษไว้เตือนจำกับตัวเองด้วยว่า ออปเจ็กต์นั้น ๆ ใช้ประโยชน์ในโปรเจ็กต์ของคุณอย่างไร (ดูรูปที่ 2)

ประโยชน์สูงสุดของออปเจ็กต์บราวเซอร์น่าจะได้แก่การเป็นทางลัดในการเรียนรู้วิธีใช้ออปเจ็กต์ต่าง ๆ ซึ่งมีอยู่ในอาณาจักรแอพพลิเคชันของไมโครซอฟต์ และหากคุณยังไม่เคยรู้จักกับวิชวลเบสิกหรือแอพพลิเคชันอื่นของไมโครซอฟต์มาก่อน ปัญหาสำคัญที่หนักอกหนักใจอันดับแรกเห็นจะเกิดขึ้นเมื่อเผชิญหน้ากับสถานการณ์เฉพาะหน้าอันไม่เป็นที่รู้แน่ชัดว่าควรใช้คอนโทรลตัวใดมาประยุกต์ หรือรู้แล้วว่าจะใช้



รูปที่ 2 หน้าต่างออปเจ็กต์บราวเซอร์หลังจากเรียกขึ้นมาด้วยปุ่ม F2

คอนโทรลไหนแต่ยังอับจนว่าควรนำพรอพเพอร์ตี้ส์หรือเมธอดใดมาใช้ดี รวมทั้งไม่ทราบจะเขียนโค้ดเข้าไปในอีเวนต์ใดเพื่อตอบรับการกระทำจากผู้ใช้อย่างถูกต้อง อันนี้ล่ะครับออปเจ็กต์บราวเซอร์ช่วยย่นเวลาได้เยอะเลย คุณแทบไม่ต้องเปิดตำราหรือคลิ๊กหาจากหน้าช่วยเหลือออนไลน์อีกต่อไป มาเล่นทางลัดกันทีเดียว

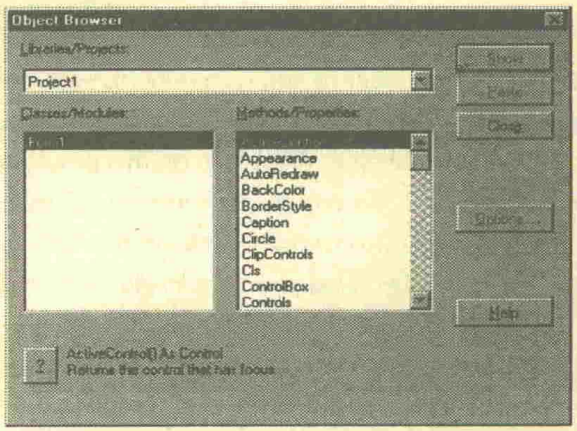
ทำความเข้าใจกันก่อนว่าหน้าต่างบราวเซอร์ในรูปที่ 2 นี้ อยู่ภายใต้โปรเจ็กต์ใหม่ซึ่งยังไม่มีการสร้างออปเจ็กต์ใด ๆ ขึ้นมาบนฟอร์ม รวมทั้งไม่มีฟอร์มหรือไฟล์อื่นนอกเหนือจาก Form1 เพียงฟอร์มเดียว โดยดีฟอลต์คุณจะเห็นในช่อง Libraries/Projects มีชื่อโปรเจ็กต์ปัจจุบันคือ Project1 อยู่ การโฟกัสที่ Project1 นี้เป็นไปเพื่อแสดงให้เห็นว่าภายใต้โปรเจ็กต์ปัจจุบันในสภาพแวดล้อมของวิชวลเบสิก ดูว่าเรามีองค์ประกอบอะไรอยู่บ้าง ก็จะเห็นในช่องแสดง Classes/Modules ว่ามีอยู่หนึ่งเดียวคือ Form1 (หากมีหลายฟอร์มหรือหลายไฟล์ชนิดอื่นก็จะเห็นปรากฏในช่องนี้ตามชื่อของพวกมัน)

หากต้องการดูว่า Form1 มีองค์ประกอบใดอยู่บ้าง คุณต้องคลิ๊กโฟกัสที่บรรทัด Form1 จะเห็นมีการแสดงรายละเอียดของฟอร์มในช่องแสดง Methods/Properties ดังที่เห็นในรูป 3

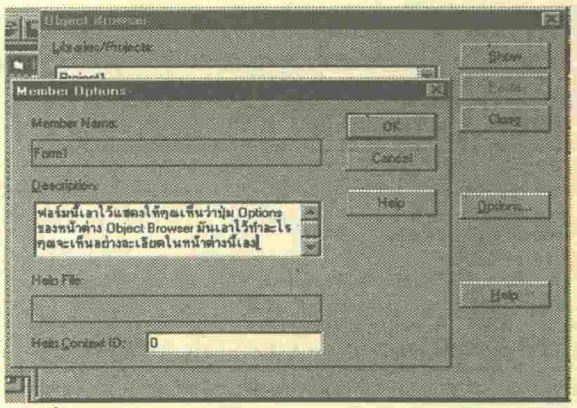
รายละเอียดต่าง ๆ นั้นก็คือ สิ่งที่จะทำให้คุณทราบว่าขอบเขตการใช้งานของฟอร์มมีอยู่เพียงใด ในที่นี้ได้แก่กลุ่มพรอพเพอร์ตี้ส์และเมธอดซึ่งนำมารวม ๆ ไว้ในที่เดียวกันโดยไม่แบ่งแยกเราเขา เมื่อคุณต้องการทราบว่าพรอพเพอร์ตี้ส์หรือเมธอดหนึ่ง ๆ มีหน้าที่อะไรเป็นหลักก็เพียงคลิ๊กโฟกัสบรรทัดนั้น เช่นในรูปที่ 3 ผมโฟกัสที่ ActiveControl เราก็จะเห็นที่ด้านล่างมุมซ้ายของหน้าต่างบราวเซอร์ประกาศว่า Active Control มีชนิดเป็น Control (พรอพเพอร์ตี้ส์อื่นอาจมีชนิดข้อมูลเป็น Integer, Boolean หรืออะไรอื่นก็จะได้รู้จากที่นี้) พร้อมทั้งกล่าวอย่างย่นย่อรวบรัดด้วยว่า ActiveControl จะคืนค่ามาเป็นคอนโทรลปัจจุบันซึ่งกำลังถูกโฟกัส หากคุณอ่านแล้วยังไม่จุใจ ไม่ใสกระจ่าง รู้สึกว่าเอาไปใช้งานไม่ได้ ก็มีปุ่ม ? ไว้ให้เรียกหน้าช่วยเหลือเฉพาะหัวข้อของ ActiveControl อีกต่างหาก

ลองคิดดูว่าเมื่อเห็นการใช้งานหลักเพียงคร่าว ๆ จะให้ประโยชน์อย่างไรบ้าง อันนี้ต้องไปเปรียบเทียบกับสมัยที่คุณเอาแต่ใช้หน้าช่วยเหลือออนไลน์ หากคุณเพียงอยากรู้ว่า ActiveControl มันทำอะไรให้ อย่างน้อยคุณต้องคลิ๊กโฟกัสฟอร์ม แล้วกดปุ่ม F1 เพื่อเรียกหน้าช่วยเหลือเฉพาะของฟอร์มขึ้นมา จากนั้นก็คลิ๊กดูพรอพเพอร์ตี้ส์ทั้งหมดของฟอร์ม แล้วเลือกคลิ๊กเฉพาะ ActiveControl จึงจะได้เห็นข้อมูลเกี่ยวกับมัน (การใช้ Search เฉพาะ ActiveControl นั้นจะได้ผลก็ต่อเมื่อคุณรู้และจำคำว่า ActiveControl ได้อยู่ก่อนแล้ว นี่เรากำลังพูดถึงสถานการณ์ทั่วไปที่เราจำชื่อเสียงเรียงนามทั้งหลายของพรอพเพอร์ตี้ส์และเมธอดในอาณาจักรวิชวลเบสิกไม่ค่อยได้)

ในการทำแอพพลิเคชันจริง คุณคงตระหนักว่าเราจำเป็นเกือบเสมอไปที่จะต้องใช้ฟอร์มหรือไฟล์อื่น เช่น .BAS มากกว่าหนึ่ง (ความ



รูปที่ 3 การโฟกัสเพื่อดูรายละเอียดของฟอร์ม



รูปที่ 4 ใช้หน้าต่าง Options ในการเขียนบันทึกช่วยจำบทบาทของฟอร์มในโปรเจ็กต์

จริงยิ่งไฟล์ในโปรเจ็กต์น้อย ๆ ล่ะก็ถือว่าดีกว่ามีไฟล์เยอะนะครับ) และลำพังชื่อที่คุณเห็นในหน้าต่างโปรเจ็กต์ก็มักไม่เพียงพอที่คุณจะเข้าใจอย่างแจ่มแจ้งว่าฟอร์มหรือไฟล์นั้น ๆ เอาไว้ทำอะไร ออปเจ็กต์บราวเซอร์อนุญาตให้คุณบันทึกช่วยเตือนความจำอย่างละเอียด เมื่อย้อนกลับมาดูอีกหรือโอนงานให้คนอื่นมารับ (เคราะห์) ต่อ ก็จะมีความสะดวกในการแกะรอยจากเส้นทางโค้ดอันทุรกันดารของคุณมากขึ้น เพราะเมื่อได้ภาพใหญ่เสียแต่ต้นมือ คือเห็นอย่างแจ่มแจ้งว่าแต่ละฟอร์มแต่ละไฟล์มีหน้าที่อะไร สอดสัมพันธ์กับองค์ประกอบอื่นอย่างไร ก้าวต่อ ๆ ไปมันก็จะทยอยมาตามลำดับเองโดยดี

เพื่อจะบันทึกช่วยจำ คุณต้องคลิ๊กโฟกัสฟอร์มที่ต้องการในลิสต์แสดง Libraries/Classes ซึ่งโปรเจ็กต์หนึ่ง ๆ มักมีอยู่หลายฟอร์ม ทันทีที่คลิ๊กโฟกัสฟอร์มใด คุณจะสังเกตเห็นปุ่ม Options ซึ่งอยู่ด้านขวามือของออปเจ็กต์บราวเซอร์มันแอคทีฟ ขอให้คลิ๊กปุ่ม Options จะเห็นหน้าต่างผุดขึ้นมาให้กำหนดอะไรต่อมิอะไรเกี่ยวกับฟอร์มได้ทันที (ดูรูปที่ 4) ซึ่งช่องใหญ่ที่เด่นที่สุดคือ Description ในช่องนี้ครับที่คุณเขียน

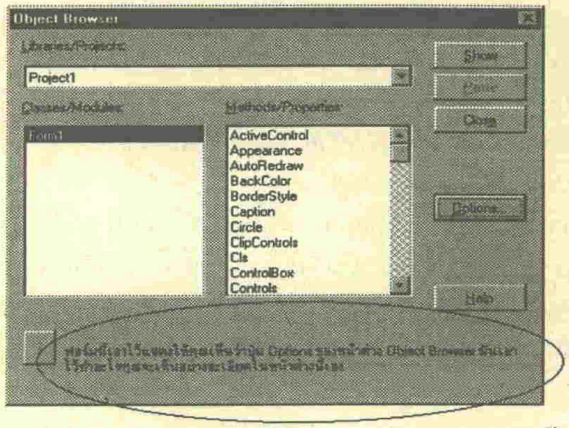
บันทึกได้โดยอิสระในรูปแบบของเวิร์ดทั่วไป ไม่จำกัดว่าจะเป็นไทยหรืออังกฤษ หากคุณเคยทำโปรเจ็กต์ที่จำเป็นจริง ๆ กับการประชุมฟอร์มหลายสิบฟอร์มพร้อมกันในโปรเจ็กต์เดียว ก็คงเห็นประโยชน์ของออปเจ็กต์บราวเซอร์ขึ้นมาอีกหน่อย เพราะหลังจากคลิ๊ก OK แล้ว นิยามที่คุณเขียนนี้จะปรากฏตรงมุมล่างซ้ายของออปเจ็กต์บราวเซอร์ทันทีดังรูปที่ 5

พอมาใช้ออปเจ็กต์บราวเซอร์ คุณสามารถเห็นการประยุกต์ใช้พรอพเพอร์ตี้ส์และเมธอดของออปเจ็กต์แต่ละตัวได้แบบสั้นกระชับ ไม่ต้องเสียเวลาเรียกหน้าช่วยเหลือเฉพาะขึ้นมาทั้งบาน (อันจะมีผลเชิงจิตวิทยาให้คุณขาดการรับรู้อย่างครอบคลุมต่อเนื่อง) เป็นการดีใช่ไหมครับถ้าคุณกำลังหาว่าจะใช้พรอพเพอร์ตี้ส์หรือเมธอดใดของออปเจ็กต์นั้น ๆ แล้วสามารถค้นหาโดยอ่านและคลิ๊กจากลิสต์ยาว ๆ คลิ๊กทีหนึ่งก็เห็นนิยามเฉพาะที่เป็นเนื้อ ๆ สั้นแค่ประโยคเดียว หากคุณรู้สึกว่าเข้าเค้าว่าน่าจะใช่สิ่งที่ต้องการ ก็ค่อยคลิ๊กเข้าไปดูรายละเอียดวิธีใช้จากหน้าช่วยเหลือเฉพาะออนไลน์ของมัน

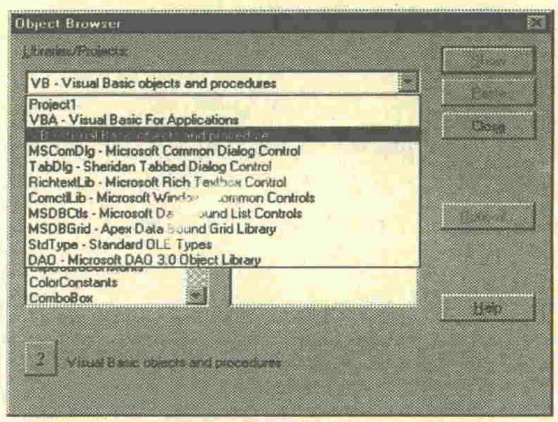
นี่ว่ากันเฉพาะประโยชน์สำหรับคนที่ยังไม่เคยรู้มาก่อน แต่ผมเห็นว่าประโยชน์สูงสุดของออปเจ็กต์บราวเซอร์น่าจะไปตกอยู่กับคนที่เคยรู้มาแล้ว ออปเจ็กต์บราวเซอร์จะเป็นเครื่องช่วยเตือนจำอันสุดวิเศษ เช่น คุณคุ้น ๆ อยู่ว่าสามารถอ้างถึงคอนโทรลซึ่งถูกโฟกัสบนฟอร์มปัจจุบันโดยไม่ต้องใช้ชื่อมันเอง แต่นึกเท่าไหร่นึกไม่ออกว่าต้องใช้พรอพเพอร์ตี้ส์หรือเมธอดใด อย่างน้อยพอจะเดาได้ล่ะว่ามันน่าจะเกี่ยวข้องกับฟอร์ม (เนื่องจากคอนโทรลเป็นองค์ประกอบย่อยของฟอร์ม) คุณก็เรียกหน้าต่างบราวเซอร์ขึ้นมาสำรวจดูว่าฟอร์มนั้นมีโครงสร้างการใช้งานอย่างไร พอเห็น ActiveControl คุณคลิ๊กแล้วก็ได้ประจักษ์ว่าใช่จริง ๆ โดยรู้จากคำอธิบายว่ามันคืนค่ามาเป็นคอนโทรลปัจจุบัน ทีนี้หากยังข้องใจเกี่ยวกับรายละเอียดวิธีใช้ ก็แล่นไปดูที่หน้าช่วยเหลือออนไลน์ของ ActiveControl ได้ด้วยปุ่ม ? ทันที

หากคุณสร้างคอนโทรลขึ้นมาบนฟอร์ม ช่องแสดง Methods/Properties จะให้เห็นเฉพาะชื่อของคอนโทรลนั้น ๆ โดยไม่มีคำอธิบายรวบรัดบอกกล่าวว่ามันใช้งานอย่างไร รวมทั้งไม่สามารถโยงไปถึงหน้าช่วยเหลือออนไลน์ของคอนโทรลนั้น ๆ อีกด้วย ที่เป็นเช่นนี้ก็เพราะช่อง Methods/Properties มุ่งแสดงองค์ประกอบของฟอร์ม หากต้องแสดงคอนโทรลก็จะเพียงแสดงชื่อเฉพาะที่เป็นพรอพเพอร์ตี้ส์ Name ของคอนโทรลเท่านั้น

คุณก็คงร้องว่าอ้าว! ถ้าอย่างนี้ออปเจ็กต์บราวเซอร์จะมีไว้ทำอะไร ดูโครงสร้างของฟอร์มได้อย่างเดียว ไม่น่าจะติดยศให้เป็นผู้สำรวจออปเจ็กต์เลย ช้าก่อนครับ ถ้าขึ้นชื่อว่าเป็นออปเจ็กต์ของไมโครซอฟต์แอพพลิเคชันแล้ว รับรองออปเจ็กต์บราวเซอร์มองเห็นหมด ไหน ๆ จะจูงใจให้ลูกค้าสร้างแอพพลิเคชันด้วยซอฟต์แวร์จากไมโครซอฟต์ทั้งยวงแล้ว ยังไงเขาก็ต้องทำให้คุณเห็นทุกอย่างถนัดโดยออกแรงน้อยที่สุด



รูปที่ 5 คุณจะเห็นคำอธิบายบทบาทของฟอร์มทันทีที่คลิ๊ก OK ออกจากหน้าต่าง Options

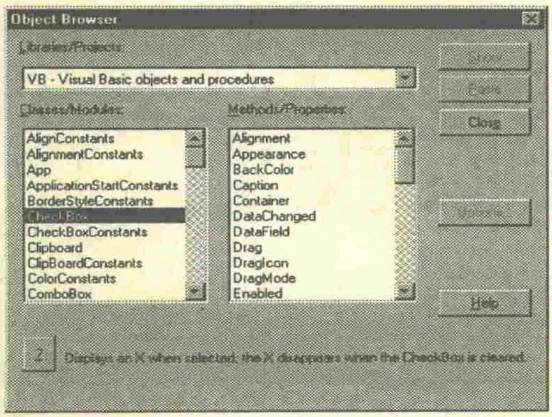


รูปที่ 6 การเลือกดูกลุ่มออปเจ็กต์ของสภาพแวดล้อมต่าง ๆ

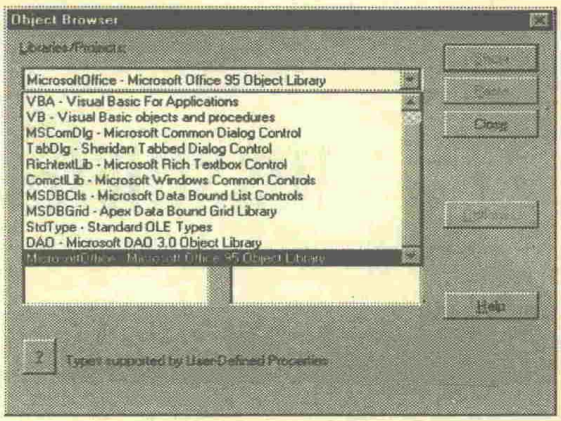
หากคุณต้องการดูออปเจ็กต์อื่นที่นอกเหนือจากฟอร์ม ไม่ว่าจะเป็นออปเจ็กต์ในสภาพแวดล้อมของวิชวลเบสิกหรือแอพพลิเคชันอื่น คุณต้องเลือกเอาจากกล่องคอมโบ Libraries/Projects ดังแสดงในลิสต์รูปที่ 6

เรามาตั้งต้นกันที่กลุ่มออปเจ็กต์ภายในสภาพแวดล้อมของวิชวลเบสิกเอง กลุ่มนี้ถูกวางตัวเป็นอันดับสามของลิสต์ เมื่อคุณคลิ๊กเลือกแล้ว จะเห็นบราวเซอร์แสดงรายการที่อยู่ในฐานะของคลาสหรือโมดูลทั้งหมดในวิชวลเบสิกให้เห็นทันที (ใครเคยใช้วิชวลเบสิก 2.0 และ 3.0 มาก่อนคงเป็นปลื้มกับออปเจ็กต์บราวเซอร์อย่างมากมายเริ่มแต่จุดนี้ไป) สมมติว่าเราสนใจ CheckBox เมื่อคลิ๊กมัน คุณจะเห็นช่องแสดง Methods/Properties มันรายงานโครงสร้างของ CheckBox ตามลำดับอักษรทันทีดังรูปที่ 7

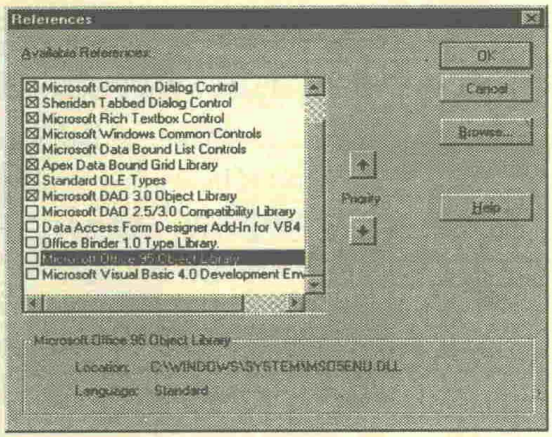
และเมื่อคุณคลิ๊กโฟกัสที่องค์ประกอบของโครงสร้างตัวใดตัวหนึ่งเช่น AddItem ก็จะเห็นคำอธิบายที่มุมล่างซ้ายของหน้าต่างบราวเซอร์ทันที ดีกว่าเปิดหน้าช่วยเหลือออนไลน์เป็นไหน ๆ



รูปที่ 7 เมื่อเลือกออปเจ็กต์ใด ก็จะเห็นโครงสร้างของออปเจ็กต์ตัวนั้นทันที



รูปที่ 9 กลุ่มออปเจ็กต์ของออฟฟิศ 95 ซึ่งเพิ่มขึ้นมาจากการเลือกในหน้าต่าง References



รูปที่ 8 หน้าต่าง References ซึ่งเราจะใช้เพิ่มรายการกลุ่มออปเจ็กต์ในช่องแสดง Libraries/Projects

ความจริงกลุ่มออปเจ็กต์ที่รวบรวมไว้ให้คุณสำรวจมีมากกว่าที่เป็นอยู่ และเพื่อจะเพิ่มกลุ่มออปเจ็กต์ที่ต้องการเข้ามาในช่องแสดง Libraries/Projects นี้ เราสามารถทำได้ด้วยหน้าต่าง Reference ซึ่งเลือกขึ้นมาจากเมนู Tools|References ดังที่เห็นในรูปที่ 8

จากรูปที่ 8 คุณจะเห็นว่าไลบรารีของออปเจ็กต์ที่อยู่ในไมโครซอฟต์ออฟฟิศ 95 ยังไม่ถูกกาเลือก ซึ่งก็ยังผลให้เรายังมองไม่เห็นกลุ่มออปเจ็กต์ของไมโครซอฟต์ออฟฟิศ 95 ในช่องแสดง Libraries/Projects ของหน้าต่างออปเจ็กต์บราวเซอร์ คราวนี้ผมลองคลิ๊กกาเลือกมัน แล้วกลับไปดูหน้าต่างออปเจ็กต์บราวเซอร์อีกที จะเห็นกลุ่มออปเจ็กต์ที่ต้องการนี้ปรากฏขึ้นมาแล้วครับ (ดูรูปที่ 9)

หากต้องการนำกลุ่มออปเจ็กต์ใดออกจากเส้นทางการสำรวจ ก็ทำได้ด้วยวิธีตรงข้ามกัน คือคลิ๊กเช็คบ็อกซ์ของไลบรารีที่ต้องการ เอากากบาทออก ง่าย ๆ เท่านี้

ก่อนจบ

ประโยชน์ที่ชัดเจนของวิชวลเบสิกที่กล่าวมาแล้วไม่ใช่ทั้งหมด หรือบางคนอาจไม่เห็นด้วย คิดว่าน่าที่ผมจะพูดถึงฟีเจอร์หลักอื่นซึ่งทรงคุณประโยชน์ยิ่งกว่านั้น ผมก็ขอกล่าวไว้ ณ ที่นี้เลยว่านี่เป็นมุมมองวิชวลเบสิก 4.0 ของผมเอง ไม่ตั้งใจจะให้คุณใช้เป็นหลักฐานอ้างอิงว่าจุดเด่นของวิชวลเบสิกคืออะไร

ยังมีฟีเจอร์อื่นมากครับที่ผมคิดจะนำมาเสนอ เช่น ความเป็นวิชวลเบสิก 4.0 ในฐานะ VBA หรือการประยุกต์คอนเสิ์ปต์เกี่ยวกับ OLE ให้เกิดประโยชน์สูงสุดในงานเฉพาะหน้า ซึ่งอันนี้เกินกำลังที่จะเขียนรวมกันไว้ทั้งหมดในตอนเดียว เพราะฉะนั้นจึงตั้งใจจะนำมาทยอยเสนอตามลำดับที่เหมาะสมต่อไป